(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 939 530 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
01.09.1999 Bulletin 1999/35

(51) Int Cl.[6]: **H04L 29/06**, H04L 12/28, H04B 1/20

(21) Application number: **99301316.8**

(22) Date of filing: **23.02.1999**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **24.02.1998 JP 4265698**
12.03.1998 JP 6170898
30.03.1998 JP 8470998

(71) Applicant: **CANON KABUSHIKI KAISHA**
Tokyo (JP)

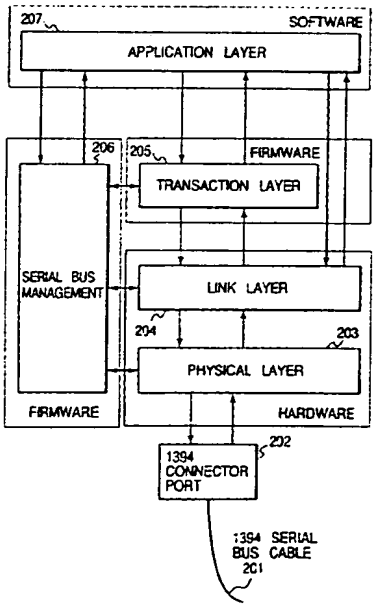
(72) Inventors:
- **Kobayashi, Takashi**
  Ohta-ku, Tokyo (JP)
- **Hatae, Shinichi**
  Ohta-ku, Tokyo (JP)
- **Niida, Mitsuo**
  Ohta-ku, Tokyo (JP)
- **Ohnishi, Shinji**
  Ohta-ku, Tokyo (JP)

(74) Representative:
Beresford, Keith Denis Lewis et al
BERESFORD & Co.
2-5 Warwick Court
High Holborn
London WC1R 5DJ (GB)

(54) Data communication system data communication method and data communication apparatus

(57) A data communication system comprises devices having at least the functions of a source, a destination, and a controller. The source transmits information data by using an address specifying one part of a memory space provided in a destination. The destination stores the information data in one part of the memory space specified by that address. The controller manages data transmission between the source and the destination. In a data communication system such as this, at least one device from among the source, the destination, and the controller resumes transmission of the information data without discarding any part of the data stored in the memory space, in the event that the transmission of the information data is interrupted in conformance with the default settings of a network.

FIG. 2



EP 0 939 530 A2

Printed by Jouve, 75001 PARIS (FR)

Description

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] This invention relates to a data communication system, data communication method and data communication apparatus, and relates in particular to a network that mixes information data (including image data) and command data and communicates said data at high speeds, and to a communication protocol that enables application of said network.

Related Background Art

[0002] Conventionally, among peripheral devices for personal computers (hereafter referred to as PCs), hard disks and printers have had the highest frequency of use. These peripheral devices were connected to the PC by means of a general-purpose digital interface such as a dedicated input/output interface or a SCSI interface (small computer system interface).

[0003] Recently, however, AV (Audio/Visual) devices such as digital cameras and digital video cameras are commanding greater interest and attention as PC peripheral devices. These AV (Audio/Visual) devices have also been connected to the PC via a dedicated interface.

[0004] With conventional dedicated interfaces and SCSI interfaces, particularly when large volumes of data are involved, such as the still images and moving images conveyed by AV devices, the rate of data transmission has been low, and because parallel communication is used, large communication cables were required, limiting the number and types of peripheral devices that could be connected to only a few devices. Numerous problems arose, such as limitations in the connection method and the inability to transmit data in real time.

[0005] The IEEE (Institute of Electrical and Electronics Engineers, Inc.) 1394-1995 standards are known as a next-generation digital interface featuring high speed and a high level of performance, and are designed to solve these problems.

[0006] Digital interfaces that conform to the IEEE 1394-1995 standards (hereafter referred to as 1394 interfaces) offer the following features.

(1) Data is transmitted at high speed.
(2) Two methods for transmitting data in real time are supported: the isochronous transmission method and the asynchronous transmission method.
(3) Connections can be configured (topology) with a high degree of freedom.
(4) Plug-and-play functions and hot-line insertion/removal functions are supported.

[0007] With IEEE 1394-1995 standards, however, certain elements such as the physical and electrical configurations of connectors and the two most fundamental data transmission methods are defined, but other elements such as the types of data, the data configuration, and the communication protocol by which data is sent and received, are not defined.

[0008] Furthermore, with the isochronous transmission method used with the IEEE 1394-1995 standards, because the response to transmitted packets is not regulated, there is no assurance as to whether each individual isochronous packet has been received. Consequently, the isochronous transmission method cannot be used to assure successful transmission of consecutive multiple data items, or of data from a single file which has been subdivided into multiple data items.

[0009] Moreover, with the isochronous transmission method used with the IEEE 1394-1995 standards, the total number of communications is restricted to 64, even if there are available transmission band widths. For this reason, it is not possible to use-the isochronous transmission method to transmit numerous communications on a small number of transmission band widths.

[0010] In addition, with the IEEE 1394-1995 standards, in cases such as the bus being reset when the power supply to a node is turned on or off, or when a node is connected or disconnected, it is necessary to interrupt the transmission of data. With the IEEE 1394-1995 standards, conversely, if data transmission is interrupted by a bus reset or an error occurring during the transmission, it is not possible to discern the contents of the data which have been lost. Furthermore, resumption of a transmission which was interrupted requires extremely complex and intricate communication procedures.

[0011] The bus reset is a function by which new topologies are recognized and the addresses assigned to the various nodes (node IDs) are set automatically. This function enables plug-and-play functions and hot-line insertion/removal functions to be supported with the IEEE 1394-1995 standards.

[0012] Additionally, with regard to communication systems conforming to the IEEE 1394-1995 standards, although real-time transmission is not required, there have been no specific proposals for a communication protocol which would enable continuous transmission of object data comprising comparatively large volumes of data (for instance, still image data, graphics data, text data, file data, and program data) where a high level of reliability is required.

SUMMARY OF THE INVENTION

[0013] An object of the present invention is to solve the above-described problems.

[0014] Another object of the invention is to provide technology which enables, in a data communication system, data communication method, and data communication apparatus, continuous and successful transmission of object data where real-time transmission is

not a necessity.

[0015] Another object of the invention is to enable the rapid resumption of data transmission, in a data communication system, data communication method, and data communication apparatus, when data transmission has been interrupted, and to provide technology that reduces the volume of data being transmitted in a redundant manner.

[0016] As a preferred embodiment for such objects, the data communication system of the present invention discloses:

a source that transmits information data by using an address specifying a part of the memory space provided in the destination;
a destination that stores the information data in a part of the memory space specified by the address; and
a controller that manages data transmission between the source and the destination;
wherein, in the event that the transmission of the information data, carried out according to a default setting of a network, is interrupted, the transmission of the information data is resumed without discarding any part of the data stored in the memory space.

[0017] As another embodiment of such objects, the data communication system of the present invention discloses:

a source that transmits information data by using an address specifying a part of the memory space provided in the destination;
a destination that stores the information data in a part of the memory space specified by the address; and
a controller that manages data transmission between the source and the destination;
wherein, in the event that the transmission of the information data, is interrupted in conformance with the default settings of a network the source resumes the transmission from one part of the information data by using an address specified by the destination or the controller.

[0018] As another embodiment of such objects, the data communication system of the present invention discloses:

a step that transmits information data using an address specifying a part of the memory space provided in the destination;
a step that stores the information data in a part of the memory space specified by the address; and
a step that resumes transmission of the information data without discarding any part of the data stored in the memory space, in the event that the transmission of the information data is interrupted according to the default setting of a network.

[0019] As another embodiment of such objects, the data communication system of the present invention discloses:

a step that transmits information data by using an address specifying a part of the memory space provided in the destination; and
a step that resumes transmission of the information data without discarding any part of the data stored in the memory space, in the event that the transmission of the information data is interrupted according to the default setting of a network.

[0020] As another embodiment of such objects, the data communication system of the present invention discloses:

a step that stores information data transmitted from a source in a part of the memory space specified by the source; and
a step that resumes transmission of the information data without discarding any part of the data stored in the memory space, in the event that the transmission of the information data is interrupted according to the default setting of a network.

[0021] As another embodiment of such objects, the data communication system of the present invention discloses:

a step that specifies the beginning of transmission of information data, along with communicating an address specifying a part of the memory space provided in the destination; and
a step that initiates control such that transmission of the information data is resumed without discarding any part of the data stored in the memory space, in the event that the transmission of the information data is interrupted according to the default settings of a network.

[0022] As another embodiment of such objects, the data communication system of the present invention discloses:

a step that transmits information data by using an address specifying a part of the memory space provided in the destination;
a step that stores the information data in a part of the memory space specified by the address; and
a step that resumes transmission of the information data from one part of the information data by using an address specified by the destination or the controller, in the event that the transmission of the information data is interrupted according to the default setting of a network.

[0023] As another embodiment of such objects, the data communication system of the present invention discloses:

a step that transmits information data by using an address specifying a part of the memory space provided in the destination; and
a step that resumes transmission of the information data from one part of the information data by using an address specified by the destination or the controller, in the event that the transmission of the information data is interrupted according to the default setting of a network.

[0024] As another embodiment of such objects, the data communication system of the present invention discloses:

a step that stores information data transmitted from a source in a part of the memory space specified by the source; and
a step that specifies an address to the source specifying one part of the memory space, in the event that the transmission of the information data is interrupted according to the default setting of a network.

[0025] As another embodiment of such objects, the data communication system of the present invention discloses:

a step that specifies the beginning of transmission of information data, along with specifying an address that specifies one part of the memory space in relation to the source; and
a step that specifies an address specifying one part of the memory space to the source, in the event that the transmission of the information data is interrupted according to the default setting of a network.

[0026] As another embodiment of such objects, the data communication system of the present invention discloses:

a unit that transmits information data by using an address specifying one part of a memory space provided in the destination; and
a unit that executes control such that transmission of the information data is resumed without discarding any part of the data stored in the memory space, in the event that the transmission of the information data is interrupted according to the default setting of a network.

[0027] As another embodiment of such objects, the data communication system of the present invention discloses:

a unit that stores information data transmitted from a source to one part of a memory space specified by the source; and
a unit that executes control such that transmission of the information data is resumed without discarding any part of the data stored in the memory space, in the event that the transmission of the information data is interrupted according to the default setting of a network.

[0028] As another embodiment of such objects, the data communication system of the present invention discloses:

a unit that transmits information data by using an address specifying one part of a memory space provided in the destination; and
a unit that initiates control such that transmission of the information data is resumed using an address specified by the destination or the controller, in the event that the transmission of the information data is interrupted according to the default setting of a network.

[0029] As another embodiment of such objects, the data communication system of the present invention discloses:

a unit that stores information data transmitted from a source to one part of a memory space specified by the source; and
a unit that specifies an address to the source that specifies one part of the memory space, in the event that the transmission of the information data is interrupted according to the default setting of a network.

[0030] Still other objects of the present invention, and the advantages thereof, will become fully apparent from the following detailed description of the embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

[0031] Fig. 1 is a block diagram indicating a configuration example of the communication system of this embodiment.

[0032] Fig. 2 is a drawing describing the configuration of the 1394 interface of this embodiment.

[0033] Fig. 3 is a drawing describing the transmission method used with the 1394 interface.

[0034] Fig. 4 is a sequence chart describing the basic procedures of the communication protocol of this embodiment.

[0035] Fig. 5 is a drawing describing a transmission model for object data.

[0036] Figs. 6A and 6B are drawings describing the address spaces of the various nodes.

[0037] Fig. 7 is a sequence chart describing the com-

munication protocol of the first embodiment.

[0038] Fig. 8 is a drawing describing the configuration of the communication packets transmitted from the controller to the source.

[0039] Fig. 9 is a drawing describing the internal addresses provided by the source.

[0040] Fig. 10 is a sequence chart describing the communication protocol of the second embodiment.

[0041] Fig. 11 is a sequence chart describing the communication protocol of the third embodiment.

[0042] Fig. 12 is a sequence chart describing the communication protocol of the fourth embodiment.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0043] The preferred embodiments of the present invention will now be described in detail hereinafter with reference to the accompanying drawings.

[0044] Fig. 1 is a block diagram describing the configuration of the communication system of this embodiment. In Fig. 1, each of devices provides a digital interface 105 conforming to the IEEE 1394-1995 standards (hereafter referred to as 1394 standards).

[0045] The communication system indicated in Fig. 1 is configured of a TV 101, a digital video tape recorder (hereafter referred to as DVTR) 102, a printer 103, and a digital camcorder (hereafter referred to as DVCR) 104.

[0046] In Fig. 1, the TV 101, DVTR 102, and DVCR 104 provide a control unit 106 and a signal processing unit 107, while the printer 103 provides a control unit 106 and an image processing unit 109. These devices are connected by means of communication cables which conform to 1394 standards.

[0047] In this embodiment, there are two types of communication cables, a 4-pin cable and a 6-pin cable. The 4-pin cable is configured of two pairs of twisted-pair shielded wires, and is used for data transmission and communication of intervention signals. The 6-pin cable is configured of two pairs of twisted-pair cables and a pair of power supply cables. Data transmitted using the two pairs of twisted-pair cables is data that has been coded using the DS-Link method.

[0048] Next, the configuration of the digital interface 105 used in this embodiment will be described in detail with reference to Fig. 2.

[0049] The digital interface 105 is functionally configured of multiple layers. In Fig. 2, the digital interface 105 is connected to the digital interfaces 105 of other devices by means of a communication cable 201 that conforms to the IEEE 1394-1995 standards. Furthermore, the digital interface 105 has at least one communication port 202, with each communication port 202 being connected to a physical layer 203 which is included in a hardware unit.

[0050] In Fig. 2, the hardware unit is configured of a physical layer 203 and a link layer 204. The physical layer 203 handles detection of physical and electrical interfaces with other nodes, bus resets and the accompanying processing, coding and decoding of input and output signals, intervention in bus usage rights, and other functions. The link layer 204 formulates communication packets, sends and receives the various types of communication packets, and handles control of cycle timers, as well as other functions. The link layer 204 also formulates the packets stipulated by the communication protocol described at a later point, and provides transmission functions.

[0051] Also, in Fig. 2, the firmware unit includes a transaction layer 205 and a serial bus management 206. The transaction layer 205 oversees the asynchronous transmission method, and provides various types of transactions (reading, writing, locking). The transaction layer 205 also provides a function by which the transactions governed by the communication protocol described at a later point are managed. The serial bus management 206 provides functions, based on the IEEE 1212 CSR standards, by which its own nodes are controlled, the connection statuses of its own nodes are managed, ID information for its own nodes is managed, and resource management of the serial bus network is carried out. The serial bus management 206 also provides a function by which control of the various types of processing operations relating to the communication protocol described at a later point is carried out.

[0052] The hardware unit and firmware unit illustrated in Fig. 2 make up, in practical terms, the 1394 interface, and the basic configuration of these units is governed by the IEEE 1394-1995 standards.

[0053] Additionally, the application layer 207 included in the software unit varies depending on the application software being used, and controls the types of object data that can be transmitted, and the manner in which they are transmitted.

[0054] The communication protocol of this embodiment, described at a later point, expands the functions of the hardware unit and firmware unit making up the digital interface 105, and provides new transmission procedures for the software unit.

[0055] With the digital interface 105 described above, when the power supply is turned on, the bus is reset automatically in response to changes in the connection configuration, such as the connection of new devices and the disconnection of devices.

[0056] A bus reset refers to processing in which the connection configuration of the various devices comprising the communication system (hereafter referred to as nodes) recognized up to the current point in time, and the communication addresses of those devices (hereafter referred to as node IDs) are initialized, the new connection configuration is recognized once again, and communication addresses are specified once again.

[0057] The next section is a brief description of the processing procedures involved in a bus reset. These procedures include recognition of the tiered connection configuration of the communication system and assign-

ment of physical communication addresses for each of the nodes.

[0058] Recognition of the connection configuration is implemented after the bus reset has begun, by a declaration of the hierarchical relationship between the various nodes. The various nodes are recognized as a communication system with a tree-type construction (tiered construction) by the nodes determining the hierarchical relationship between each of the nodes. Because the hierarchical relationship between each of the nodes depends on the connection status of the communication system and on the functions of the various nodes, the same relationship does not necessarily result each time a bus reset is carried out.

[0059] For instance, in the communication system illustrated by Fig. 1, first, the various digital interfaces 105 determine the hierarchical relationship between the printer 103 (hereafter referred to as Node D) and the DVTR 102 (hereafter referred to as Node C). Next, the hierarchical relationships between the DVCR 104 (hereafter referred to as Node B) and the TV 101 (hereafter referred to as Node A), and between Node C and Node A, are specified.

[0060] Finally, all of the devices recognized as having nodes with a higher hierarchical (or upper-level) position serve as root nodes, and management of intervention in the bus usage rights of this communication system is carried out. In the communication system illustrated in Fig. 1, Node A is the root node.

[0061] After the root nodes have been determined, specification of node IDs for the various nodes configuring the communication system begins automatically. Basically, node IDs are specified by the node with the higher hierarchical position allowing a physical address to be specified for a node with a lower hierarchical position which is connected to a communication port with a lower port number, following which the lower-position node grants permission for its own subsidiary nodes to specify addresses, and the processing continues in sequential order. Nodes for which their own node IDs have already been specified transmit their own ID packets, and notify other nodes of the node IDs assigned to them. Ultimately, after node IDs have been specified for all of the lower-position nodes, the higher-position nodes specify their own node IDs.

[0062] In the processing described above, as processing is carried out repeatedly, the node IDs for the root nodes are the last to be specified. Because the node IDs assigned to the various nodes depend on the hierarchical relationship between the various devices, the same node ID will not necessarily be specified for the same node each time a bus reset occurs.

[0063] Next, the process by which node IDs are automatically specified will be described in reference to Fig. 1. The following description will apply to a case in which, after the connection configuration has been recognized, Node A serves as the root node.

[0064] In Fig. 1, Node A, which is the root node, first allows a node ID to be specified for the node connected to the "Port 1" communication port, namely, Node B.

[0065] Node B specifies its own node ID as "#0", and broadcasts the resulting address to all of the nodes configuring the communication system, as a self-ID packet. Here, "broadcast" refers to specific information being sent to the addresses of numerous unspecified nodes.

[0066] As a result, all of the nodes acknowledge that the node ID "#0" has already been assigned, and the next node for which address specification is permitted specifies the next node address as "#1". After Node B has been specified, Node A allows specification of a node ID for the node connected to the "Port 2" communication port, namely Node C.

[0067] Node C grants permission for IDs to be specified for the communication ports connected to the lower-position nodes, in sequential order, starting with the lowest port number. In other words, permission is granted for Node D, and after a node ID of "#1" is specified for Node D, and Node D, having accepted that permission, then broadcasts a self-ID packet.

[0068] After Node D has been specified, Node C specifies its own node ID as "#3", and finally, Node A, which is the root node, specifies its own node ID as "#4", and recognition of the connection configuration is concluded.

[0069] Through this type of bus reset processing, the digital interfaces 105 are able to automatically recognize the connection configuration of the communication system and specify the communication addresses for the various nodes. The various nodes, using the node IDs described above, are then able to carry out communications among themselves.

[0070] Next, the data transmission method provided by the digital interfaces 105 will be described in reference to Fig. 3.

[0071] The communication system illustrated in Fig. 3 provides two data transmission methods, the isochronous transmission mode and the asynchronous transmission mode. Because the isochronous transmission mode assures dispatch and reception of a packet containing a given quantity of data within the transmission cycle for one packet (125 μs), this mode is effective for real-time transmission of video data and audio data. The asynchronous transmission mode sends and receives control commands, file data, and other data in an asynchronous manner, as necessary, and the priority order specified for this mode is lower than that of the isochronous transmission mode.

[0072] In Fig. 3, at the beginning of each communication cycle, a communication packet called a cycle start packet 301 is dispatched, which adjusts the cycle times for the timings of the various nodes.

[0073] After the cycle start packet 301 has been dispatched, the isochronous transmission mode is specified for a given period of time. In the isochronous transmission mode, multiple isochronous transmissions can be carried out by assigning channel numbers to the re-

spective data items being transmitted based on the isochronous transmission mode.

[0074] For instance, in Fig. 3, if a channel number of "ch O" is assigned to the data 302 being dispatched via isochronous transmission from the DVCR 104, while a channel number of "ch 1" is assigned to the data 303 being dispatched via isochronous transmission from the DVTR 102, and a channel number of "ch 2" is assigned to the data 304 being dispatched via isochronous transmission from the TV101, the various data items are transmitted using isochronous transmission, within a single communication cycle, at the appropriate timing.

[0075] When the various isochronous transmissions have been completed, asynchronous transmission is used until the start cycle packet 301 for the next cycle is transmitted. For instance, in Fig. 3, data 305 is sent from the DVCR 104 to the printer 103, based on asynchronous transmission.

[0076] Fig. 4 shows a sequence chart describing the basic configuration of the communication protocol of this embodiment, using the asynchronous transmission mode. In Fig. 4, the object data (for example, still image data) sequence is sent to the node which handles asynchronous transmission, namely source 402, which is the DVCR 104. Then, the object data sent in sequential order from source 402, using asynchronous transmission, is sent to the reception node, namely the destination 403, which is the printer 103. Following that, the TV 101 is set as the node which controls communications between the source 402 and the destination 403, namely, the controller 401.

[0077] The communication protocol for this embodiment consists of three phases. The first phase 404 is the connection phase, when the controller 401 inquires whether or not the destination 403 has a reception buffer large enough to contain the destination offset, which will be described at a later point, and the destination 403 is set in the reception standby state. The controller 401 selects the object data to be sent from the source 402 using asynchronous transmission, and sets the transmission from the transmission buffer.

[0078] The second phase 405 is the transmission phase, in which the controller 401 controls the source 402 and the destination 403, and transmits the object data in sequential order, by means of at least one packet, using asynchronous transmission.

[0079] The third phase 406 is the connection release phase, in which the controller 401 releases the reception buffer of the destination 403 from its own management, and in the same way, releases the transmission buffer of the source 402 from its management.

[0080] Fig. 5 is a drawing which describes the relationship between the object data sent from the source 402 using asynchronous transmission and the reception buffer of the destination 403.

[0081] The object data 501 sent from the source 402 using asynchronous transmission is divided into at least one segmented data item 502 which is equivalent to the size of the reception buffer of the destination 403 communicated by the controller 401.

[0082] The individual segmented data items 502 are organized into one or more communication packets 503 based on the asynchronous transmission mode (hereafter called asynchronous packets 503), and are sent in sequential order from the source 402 to the destination 403.

[0083] The destination 403 receives the asynchronous packets 503 sent in sequential order from the source 402, and writes them temporarily to a reception buffer 504. After transmission of one segment of object data has been completed, the destination 403 writes the individual segments of data stored in the reception buffer 504 to the internal memory 505, in sequential order.

[0084] Next, the reception buffer 504, possessed by all of the nodes, including the destination 403, will be described in detail with reference to Figs. 6A and 6B. The reception buffer 504 is managed by a 64-bit address space conforming to IEEE 1212 CSR (Control and Status Register Architecture) standards (or to ISO/IEC 13213: 1994 standards). IEEE 1212 CSR standards are standards which govern control, management, and address assignments for serial buses.

[0085] Fig. 6A shows a logical memory space indicated by a 64-bit address. Fig. 6B shows one part of the address space illustrated by Fig. 6A; for instance, the address space in which the most significant 16 bits will serve as $FFFF_{16}$. The reception buffer 504 uses part of the address space illustrated by Fig. 6B, and is specified by the destination offset indicated by the least significant 48 bits of the address. These destination offsets are specified by the header sections of the various asynchronous packets.

[0086] In Fig. 6B, for instance, 000000000000_{16} to $0000000003FF_{16}$ are reserved fields, and the fields in which the object data 501 is actually written are the 48 least significant bits of the address, expressed by $FFFFF0000400_{16}$ and subsequent bits.

(First embodiment)

[0087] The communication protocol of the first embodiment is described as it applies to the communication system illustrated by Fig. 1. Here, the TV 101 serves as the controller 401 of the first embodiment, the DVCR 104 as the source 402, and the printer 103 as the destination 403.

[0088] In the first embodiment, after the object data (for example, image data, audio data, graphics data, or text data) has been divided into segments consisting of one or more data, as illustrated in Fig. 5, the source 402 sends the segmented data in the form of one or more asynchronous packets, using asynchronous transmission. The destination 403 writes the one or more asynchronous packets sent from the source 402 using asynchronous transmission to the reception buffer 504, and stores the data in the internal memory 505 in units of

one segment of data.

[0089] The controller 401 selects object data sent from the source 402 in conjunction with the size of the destination 403 buffer, and manages communications with the source 402 involving specifications such as the size of the segmented data formulated by the source 402, and other parameters.

[0090] The communication protocol with the configuration of the first embodiment includes, as shown in Fig. 4, three phases, namely a connection phase, a transmission phase, and a connection release phase.

[0091] The connection phase and connection release phase of the configuration of the first embodiment can be executed in the same way as the first phase 404 and the third phase 406 shown in Fig. 4. Consequently, with regard to the configuration of the first embodiment, the transmission phase is described in detail.

[0092] Fig. 7 is a sequence chart describing the transmission phase of the configuration of the first embodiment in detail.

[0093] In Fig. 7, the controller 401 specifies (704) to the destination 403 that the segmented data sent by means of asynchronous transmission is to be received in a number of communication packets. The controller 401 also specifies (705) to the source 402 that the object data is to be divided into one or more segmented data, and that these segmented data are to be transmitted by means of asynchronous transmission in a number of communication packets.

[0094] The configuration of the data section of the communication packet sent from the controller 401 to the source 402 is described here with reference to Fig. 8. The communication packet shown in Fig. 8 is a communication packet which instructs the source 402 to begin transmitting the segmented data, and the data is sent using the asynchronous transmission mode. The communication packet shown in Fig. 8 has a horizontal width of eight bytes.

[0095] In Fig. 8, the command which instructs the transmission is stored in the first byte field 801. The segment number indicating the sequential order in which segmented data is to be transmitted is stored in field 802, and the node ID information for the destination 403 is stored in field 803.

[0096] The initial address of the reception buffer 504 provided by the destination 403 is stored in field 804. Field 805 contains information pertaining to the size of the reception buffer provided by the destination 403, meaning the size of one data segment. Field 806 contains information pertaining to the maximum size of communication packets that can be received by the destination 403. Field 807 contains various types of status information.

[0097] In Fig. 8, an area 808 which contains the re-send identification bit is specified in the area designated for field 807. The source 402 reads this area 808 and determines whether normal transmission processing or re-send processing is to be carried out. For example, if the controller 401 indicates that normal segmented data is to be transmitted from the source 402, this area 808 will contain a "0".

[0098] In the configuration of the first embodiment, the source 402 which receives the communication packet shown in Fig. 8 is configured so that the value for the initial address of the reception buffer 504 stored in field 804 is stored in a designated internal register. Here, the internal register is included in the digital interfaces 105 provided by the various devices or the control unit 106 which controls the operations of the various devices (see Fig. 1).

[0099] In Fig. 4, following the instruction from the controller 401, the source 402 organizes single data segments into one or more asynchronous packets, and sends these asynchronous packets in sequential order (706) to the destination 403. For instance, as shown in Fig. 5, the source 402 divides single-segment data into n data, and after configuring n asynchronous packets from these data, sends them in sequential order.

[0100] Here, the addresses (destination offset) specified by the designated field of the reception buffer 504 provided by the destination 403 are stored in the various asynchronous packets. For instance, the first asynchronous packet of the segmented data contains the initial address of the reception buffer 504, as communicated by the controller 401. Subsequent asynchronous packets contain the offset addresses specifying the designated fields of the reception buffer 504 in sequential order. If a bus reset 707 occurs during asynchronous transmission of single-segment data, the source 402 interrupts the transmission of the segmented data.

[0101] The destination 403, along with interrupting the reception of the segmented data, stores the offset address included in the last asynchronous packet to be received normally before the bus reset occurred in the internal register described above. The destination 403 also retains the segmented data stored in the reception buffer 504 without discarding any part of the data.

[0102] After bus reset processing has been concluded, the controller 401 which detected the bus reset 707 checks to see whether or not the node IDs of the source 402 and the destination 403 have been changed. Following that, the controller 401 issues instructions to the source 402 and the destination 403 to resume data transmission (708 and 709).

[0103] The communication packet carrying the instruction to resume transmission which is sent to the source 402 contains the node ID of the destination 403 which was specified by the bus reset 707. In addition, the data in this communication packet is configured as shown in Fig. 8, and a "1" is stored in area 808 (the re-send identification bit) inside field 807 in that communication packet. The source 402 reads this area 808 and recognizes that re-send processing is to be carried out.

[0104] In the same way, the communication packet which instructs the destination 403 that transmission is to be resumed contains the node ID of the source 402

specified by the bus reset 707.

[0105] The destination 403, which received the instruction from the controller 401, notifies the source 402 of the offset address stored in the internal register, and also waits (710) for the segmented data to be re-sent from the source 402.

[0106] The source 402, which received the instruction from the controller 401, waits for notification from the destination 403 of the offset address. Following notification of the offset address, the source resumes asynchronous transmission of the segmented data from partway through the data (711).

[0107] At this point, the source 402 compares the value of the initial address of the reception buffer 504 provided by the destination 403 with the value of the offset address communicated by that destination 403, and identifies the portion of the segmented data from which transmission is to resume, based on the difference between the two values.

[0108] For example, if the least significant 16th bit of the initial address of the reception buffer 504 stored in the internal register is "OEOOh", and the least significant 16th bit of the offset address described above is "OE04h", as shown in Fig. 9, the source 402 will resume asynchronous transmission from the data 901 of the fifth byte of the segmented data being transmitted when the transmission was interrupted by the bus reset.

[0109] After asynchronous transmission of this single-segment data has been completed, the source 402 reports (712) to the controller 401 that the transmission has been completed. In the same way, the destination 403 reports (713) to the controller 401 that reception of a single-segment data has been completed.

[0110] In this way, even if a bus reset occurs during transmission of a single-segment data, executing the procedure indicated by 704 to 713 in Fig. 7 enables transmission of that entire segment of data to be resumed, without having to begin again from the beginning. Also, to begin transmission from the next or subsequent data segments, the controller 401, source 402, and destination 403 need only repeat the procedures indicated by 704 to 713.

[0111] As described above, in the configuration of the first embodiment, after the controller 401 detects a bus reset, the source 402 and destination 403 are requested to resume transmission. The source 402, using the offset address communicated by the destination 403 which received the request for resumed transmission, is configured in such a way that it selects the data segments which need to be sent, and sends the data in sequential order, using asynchronous transmission.

[0112] Through this procedure, even if a bus reset occurs while segmented data is being transmitted, the time required to restart the transmission and the volume of data caused by redundant transmission can be reduced, at the same time avoiding a deterioration in the transmission efficiency.

(Configuration of the second embodiment)

[0113] In the following section, the communication protocol of the configuration of the second embodiment will be described, applying the communication system shown in Fig. 1, in the same way as for the configuration of the first embodiment. In this description, for the configuration of the second embodiment, the controller 401 is the TV 101, the source 402 is the DVCR 104, and the destination 403 is the printer 103.

[0114] Subsequently, in the configuration of the second embodiment, descriptions of those materials or functions which are identical to or equivalent to those of the configuration of the first embodiment will be omitted, using the same symbols as those used in the configuration of the first embodiment.

[0115] The communication protocol of the configuration of the second embodiment, like that of the first embodiment, comprises, as shown in Fig. 4, three phases, namely, a connection phase 404, a transmission phase 405 and a connection release phase 406. Subsequently, in the configuration of the second embodiment, the transmission phase will be described in detail, in the same way as that of the first embodiment.

[0116] Fig. 10 is a sequence chart describing the transmission phase of the configuration of the second embodiment in detail.

[0117] In Fig. 10, because the procedure up to the point where a bus reset 707 occurs, meaning the procedure consisting of steps 704 to 706 in Fig. 7, is the same as that for the configuration of the first embodiment, a description of that procedure is omitted. The following section describes the processing which takes place following occurrence of a bus reset 707.

[0118] If a bus reset 707 occurs during asynchronous transmission of single-segment data, the source 402 interrupts the transmission of the segmented data. Along with interrupting the transmission of the segmented data, the source 402 stores the offset address which is included in the last asynchronous packet to be received normally before the bus reset occurred in an internal register.

[0119] The internal register in which the offset address is stored is included in the digital interfaces 105 provided by the various devices or the control unit 106 that controls the operations of the various devices. The offset address described above is stored in an internal register which is different from that in which the initial address of the reception buffer is stored.

[0120] The destination 403, along with interrupting reception of the segmented data, retains the segmented data stored in the reception buffer without discarding any part of the data.

[0121] After bus reset processing has been concluded, the controller 401 which detected the bus reset 707 checks to see whether or not the node IDs of the source 402 and the destination 403 have been changed. Following that, the controller 401 issues instructions to the

source 402 and the destination 403 to resume data transmission (1001 and 1002).

[0122] The communication packet carrying the instruction to resume transmission which is sent to the source 402 contains the node ID of the destination 403 which was specified by the bus reset 707. In addition, the data in this communication packet is configured as shown in Fig. 8, and a "1" is stored in area 808 (the re-send identification bit) inside field 807 in that communication packet. The source 402 reads this area 808 and recognizes that re-send processing is to be carried out.

[0123] In the same way, the communication packet which instructs the destination 403 that transmission is to be resumed contains the node ID of the source 402 specified by the bus reset 707.

[0124] The destination 403, which received the instruction from the controller 401, waits for the segmented data to be re-sent from the source 402. The source 402, which received the instruction from the controller 401, reads the offset address stored in the internal register described above, and resumes asynchronous transmission starting from the data corresponding to that address (1003).

[0125] At this point, the source 402 compares the value of the initial address of the reception buffer 504 provided by the destination 403 with the value of the offset address stored in the internal register described above, and identifies the portion of the segmented data from which transmission is to resume, based on the difference between the two values.

[0126] For example, if the least significant 16th bit of the initial address of the reception buffer stored in the internal register is "OEOOh", and the least significant 16th bit of the offset address described above is "OE04h", as shown in Fig. 9, the source 402 will resume asynchronous transmission from the data 901 of the fifth byte of the segmented data being transmitted when the transmission was interrupted by the bus reset.

[0127] After asynchronous transmission of this single-segment data has been completed, the source 402 reports (712) to the controller 401 that the transmission has been completed. In the same way, the destination 403 reports (713) to the controller 401 that reception of a single-segment data has been completed.

[0128] In this way, even if a bus reset occurs during transmission of a single-segment data, executing the procedure indicated in Fig. 10 enables transmission of that entire segment of data to be resumed, without having to begin again from the beginning.

[0129] Also, to begin transmission from the next or subsequent data segments, the controller 401, source 402, and destination 403 need only repeat the procedure indicated in Fig. 10.

[0130] As described above, in the configuration of the second embodiment, after the controller 401 detects a bus reset, the source 402 and destination 403 are requested to resume transmission. The source 402 is configured in such a way that it uses the offset address included in the asynchronous packet transmitted normally prior to the bus reset, selects the portion of the segmented data which needs to be sent, and sends the data in sequential order using asynchronous transmission.

[0131] Through this procedure, even if a bus reset occurs while segmented data is being transmitted, in the same way as the configuration of the first embodiment, the time required to restart the transmission and the volume of data caused by redundant transmission can be reduced, at the same time avoiding a deterioration in the transmission efficiency.

(Configuration of the third embodiment)

[0132] In the following section, the communication protocol of the configuration of the third embodiment will be described, applying the communication system shown in Fig. 1. In this description, for the configuration of the third embodiment, the controller 401 is the TV 101, the source 402 is the DVCR 104, and the destination 403 is the printer 103.

[0133] Subsequently, in the configuration of the third embodiment, descriptions of those materials or functions which are identical to or equivalent to those of the configuration of the first embodiment will be omitted, using the same symbols as those used in the configuration of the first embodiment.

[0134] In the configuration of the third embodiment, after the information data has been divided into one or more data segments, the source 402 transmits those data segments as one or more asynchronous packets, using asynchronous transmission.

[0135] The destination 403 receives the one or more asynchronous packets sent from the source 402 using asynchronous transmission, and stores the data in an internal memory, in single data segments. The controller 401 selects information data sent from the source 402 in conjunction with the size of the destination 403 buffer, and manages communications with the source 402 involving specifications such as the size of the segmented data formulated by the source 402, and other parameters.

[0136] The communication protocol of the configuration of the third embodiment, like that of the first embodiment, comprises three phases, namely, a connection phase 404, a transmission phase 405 and a connection release phase 406. Subsequently, in the configuration of the third embodiment, the transmission phase will be described in detail.

[0137] Fig. 11 is a sequence chart describing the transmission phase of the configuration of the third embodiment in detail.

[0138] In Fig. 11, because the procedure up to the point where a bus reset 707 occurs, meaning the procedure consisting of steps 1104 to 1106 in Fig. 11, is the same as that for the configuration of the first embodiment, a description of that procedure is omitted. The following section describes the processing which takes

place following occurrence of a bus reset 707.

[0139] If a bus reset 707 occurs during asynchronous transmission of single-segment data, the source 402 interrupts the transmission of the segmented data. Along with interrupting the transmission of the segmented data, the source 402 stores the offset address which is included in the last asynchronous packet to be received normally before the bus reset occurred in an internal register.

[0140] The internal register in which the offset address is stored is included in the digital interfaces 105 provided by the various devices or the control unit 106 that controls the operations of the various devices. The offset address described above is stored in an internal register which is different from that in which the initial address of the reception buffer is stored.

[0141] The destination 403, in addition to interrupting reception of the segmented data, also retains the segmented data stored in the reception buffer without discarding any part of the data.

[0142] After bus reset processing has been concluded, the controller 401 which detected the bus reset 707 checks to see whether or not the node IDs of the source 402 and the destination 403 have been changed. Following that, the controller 401 requests from the source 402 the offset address containing the last asynchronous packet to have been transmitted normally (1101).

[0143] The data in the communication packet sent to the source 402 is configured as shown in Fig. 8, and a "1" is stored in area 808 (the re-send identification bit) inside field 807 in that communication packet. The source 402 reads this area 808 and recognizes that re-send processing is to be carried out.

[0144] In response to this request, the source 402 reads the offset address stored in the internal register described above, and communicates the address to the controller 401 (1102). The controller 401, after receiving the notification from the source 402, issues an instruction (1103) to resume asynchronous transmission starting from the data corresponding to that offset address.

[0145] The source 402, which received the instruction from the controller 401 to resume transmission, resumes asynchronous transmission starting from the data corresponding to the offset address described above (1104).

[0146] At this point, the source 402 compares the value of the initial address of the reception buffer 504 provided by the destination 403 with the value of the offset address described above, and identifies the portion of the segmented data from which transmission is to resume, based on the difference between the two values.

[0147] For example, if the least significant 16th bit of the initial address of the reception buffer 504 stored in the internal register is "OEOOh", and the least significant 16th bit of the offset address described above is "OE04h", as shown in Fig. 9, the source 402 will resume asynchronous transmission from the data 901 of the fifth byte of the segmented data being transmitted when the transmission was interrupted by the bus reset.

[0148] After asynchronous transmission of this single-segment data has been completed, the source 402 reports (712) to the controller 401 that the transmission has been completed. In the same way, the destination 403 reports (713) to the controller 401 that reception of a single-segment data has been completed.

[0149] In this way, even if a bus reset occurs during transmission of a single-segment data, executing the procedure indicated in Fig. 11 enables transmission of that entire segment of data to be resumed, without having to begin again from the beginning.

[0150] Also, to begin transmission from the next or subsequent data segments, the controller 401, source 402, and destination 403 need only repeat the procedure indicated in Fig. 11.

[0151] As described above, in the configuration of the third embodiment, after the controller 401 detects a bus reset, it asks the source 402 for the offset address included in the asynchronous packet transmitted normally prior to the bus reset. The controller 401 is configured so that it then instructs the source 402 to resume asynchronous transmission starting from the data segment corresponding to the offset address communicated from the source 402.

[0152] Through this procedure, even if a bus reset occurs while segmented data is being transmitted, in the same way as the configuration of the first embodiment, the time required to restart the transmission and the volume of data caused by redundant transmission can be reduced, at the same time avoiding a deterioration in the transmission efficiency.

[Configuration of the embodiment of the invention]

(Fourth embodiment)

[0153] The following section describes the configuration of the fourth embodiment, based on the drawings.

[0154] The flow of the data transmission phase 405 of the present invention is shown in Fig. 13. The connection phase 404 and the connection release phase 406 are the same as those shown in Fig. 4.

[0155] In Fig. 13, the controller 401 first issues an instruction (1004) to the destination 403 to receive the segmented data, and instructs the source 402 to send the segmented data (1205). The field 804 of the command packet issued here by the controller 401 serves as the initial address of the buffer 504 of the destination 403. Also, the re-send identification bit 808 in the status field 807 is set to "0", which indicates that the command is an ordinary segment transmission command.

[0156] The source 402 stores the value for the initial address 804 of the buffer 504 of the destination 403 in an internal register. The various segments of data are sent (1206) from the source 402 to the destination 403 as a number of asynchronous packets.

[0157] Assuming a bus reset (1207) occurring at this point, the source 402 interrupts the data transmission. The destination 403, along with interrupting the reception of the data, stores the offset address which is included in the last asynchronous packet to be received normally before the bus reset occurred in an internal register. The data in the destination 403 buffer 504 is retained, without any part of the data being discarded.

[0158] Following this, the controller 401 which detected the bus reset requests from the destination 403 the offset address of the last asynchronous packet to have been received normally (1208). The destination 403 notifies (1209) the controller 401 of the offset address stored in the internal register.

[0159] The controller 401 notifies the source 402 of the offset address communicated from the destination 403, and also instructs that transmission is to be resumed (1210). At this time, the value of the re-send identification bit 805 in the status field 807 of the command sent from the controller 401 is "1", indicating that the command is an instruction to send the segment transmission once again. The field 804 also serves as the offset address of the buffer which was communicated from the destination 403.

[0160] The source 402 selects the data in the segment from which the re-transmission is to begin. This is done by identifying the difference between the initial address of the reception buffer 504 of the destination 403, which was stored in an internal register when the first re-send command was received, and the value in the re-send command field 804.

[0161] For example, if the least significant 16th bit of the initial address of the reception buffer of the destination 403 stored in the internal register when the first re-send command was received is "OEOOh", and the least significant 16th bit of the offset address in the re-send command field 808 is "OE04h", as shown in Fig. 9, the source 402 will resume asynchronous transmission from the data 901 of the fifth byte of the segmented data being transmitted when the transmission was interrupted by the bus reset (1215).

[0162] After asynchronous transmission of this segmented data has been completed, normally the source 402 reports (1211) to the controller 401 that the transmission has been completed. The destination 403 also reports (1212) to the controller 401 that reception of the segmented data has been completed.

[0163] When the transmission of the segmented data which was interrupted has been completed, the controller 401 instructs the destination 403 once again to receive the next segmented data (1213), and instructs the source 402 to send the next segmented data (1214). At this point, the command field 804 serves as the initial address of the buffer for the destination 403, and the value of the re-send identification bit 808 in the status field 807 is set to "0", which indicates that the command is an ordinary segment transmission command.

[0164] By configuring the data communication system of the present embodiment in this manner, segments can be transmitted without starting over from the beginning even if a bus reset occurs, and unnecessary retransmission of data can be eliminated. In addition, there is no need to reduce the size of the destination buffer, so that communication is kept to the segmented data itself, suppressing increases in the volume of bus traffic and improving transmission efficiency.

(Configuration of the fifth embodiment)

[0165] The configurations of the embodiments 1 to 4 described above can be actualized by means of software. For example, these embodiments are configured in such a way that a storage medium on which software program codes which actualize the functions of the embodiment configurations described above have been stored can be supplied to control units (including microcomputers) provided by the controller 401, source 402, and destination 403 of the present embodiment configurations.

[0166] The embodiment configurations of the present invention can be actualized by the control units provided by the controller 401, source 402, and destination 403 of the present embodiment configurations reading the program codes stored on the storage medium, and controlling the operations of the system or the device so as to achieve the functions of the embodiment configurations described above.

[0167] For example, a storage medium containing program codes which actualize the processing and functions of the various embodiments shown in Figs. 7, 10, 11, and 12 is supplied to the control units 106 of the nodes which serve as the controller 401, source 402 and destination 403.

[0168] The control units 106 of the nodes which serve as the controller 401, source 402 and destination 403 may then read the program codes stored on the storage medium, and operate so as to actualize the functions of the various embodiment configurations.

[0169] In this case, the program codes read from the storage medium actualize the functions of the embodiment configurations described above themselves, which makes the storage medium on which those program codes have been stored a configuration element of the present invention.

[0170] As a storage medium from which the program codes are supplied, various media may be used, such as, for example, a floppy disk, hard disk, optical disk, optomagnetic disk, CD-ROM, CD-R, magnetic tape, non-volatile memory card, or ROM.

[0171] In addition, a medium such as an OS (operating system) or application software running in the control unit, based on the instructions of the program codes read from the storage medium, might be included in the present invention to actualize the functions of the embodiment configurations described above, by controlling the operations of the system or devices of the configu-

ration of the present embodiment.

[0172] Furthermore, after the program codes read from the storage medium have been written to a function expansion board connected to a control unit or a memory provided by a function expansion unit, the control unit provided by said function expansion board may be said to be included in the present invention if the functions of the embodiment configurations described above are actualized by controlling the operation of the system or devices of the present embodiment configuration.

[0173] Execution may be carried out in various manners without straying from the spirit or the primary features of the present invention.

[0174] For example, in the configurations of the first to fourth embodiments, the functions of the controller 401, the source 402 and the destination 403 were described assuming that those various devices were provided, but the present invention is not necessarily limited to those circumstances. For instance, the configuration might be designed to include devices which provide the same functions as those of the controller 401 and the source 402 indicated in the configurations of the first to fourth embodiments.

[0175] If the present invention is configured in this way, because communication between the controller 401 and the source 402 is carried out without a transmission channel being shared by the various devices, it may be possible to increase the transmission efficiency of the overall communication system.

[0176] Consequently, the embodiment configurations described above are no more than examples of a number of points, and should not be interpreted in a limiting manner.

Claims

1. A data communication system comprising: a source for transmitting information data by using an address specifying a part of the memory space provided in a destination;

   a destination for storing the information data in a part of the memory space specified by the address; and
   a controller for managing data transmission between said source and said destination;
   wherein, when a transmission of the information data is interrupted according to a default setting of a network, the transmission of the information data is resumed without discarding any part of the data stored in said memory space.

2. A data communication system according to claim 1, wherein one or more segments of said information data are transmitted in one or more packets.

3. A data communication system according to claim 2, wherein said packets are transmitted continuously.

4. A data communication system according to claims 2 and 3, wherein said address is stored in the packets.

5. A data communication system according to claims 2 to 4, wherein said address is different in each of the packets.

6. A data communication system according to claims 1 to 5, wherein the transmission of the information data is resumed by using an address specified by said controller.

7. A data communication system according to claims 1 to 5, wherein the transmission of said information data is resumed by using an address specified by said destination.

8. A data communication system according to claims 1 to 5, wherein the transmission of said information data is resumed by using an address managed by said source.

9. A data communication system according to claims 1 to 8, wherein said default setting is carried out when the connection configuration of said network has been changed.

10. A data communication system according to claims 1 to 9, wherein said default setting includes a proces that automatically recognizes a connection configuration of said network.

11. A data communication system according to claims 1 to 10, wherein said information data is transmitted by using the asynchronous transmission method of an IEEE 1394-1995 standards.

12. A data communication system according to claims 1 to 11, wherein said network is a network which conforms to an IEEE 1394-1995 standards.

13. A data communication system according to claims 1 to 12, wherein said information data is at least one of image data, graphics data or text data.

14. A data communication system comprising: a source for transmitting information data by using an address specifying a part of the memory space provided in a destination;

    a destination for storing the information data in a part of the memory space specified by the address; and
    a controller for managing data transmission be-

tween said source and said destination;
wherein, when a transmission of the information data is interrupted according to a default setting of a network, said source resumes the transmission from one part of the information data by using an address specified by said destination or said controller.

15. A data communication method comprising the steps of:

transmitting information data by using an address specifying a part of a memory space provided in a destination;
storing said information data in a part of the memory space specified by said address; and
resuming a transmission of said information data without discarding any part of the data stored in the memory space, when the transmission of the information data is interrupted according to a default setting of a network.

16. A data communication method comprising the steps of:

transmitting information data by using an address specifying a part of a memory space provided in a destination; and
resuming a transmission of the information data without discarding any part of the data stored in the memory space, when the transmission of said information data is interrupted according to a default setting of a network.

17. A data communication method comprising the steps of:

storing information data transmitted from a source in a part of the memory space provided in a relevant source; and
resuming a transmission of the information data without discarding any part of the data stored in the memory space, when the transmission of said information data is interrupted according to a default setting of a network.

18. A data communication method comprising the steps of:

specifying a beginning of transmission of information data, along with communicating an address specifying a part of a memory space provided in a destination; and
controlling to resume the transmission of the information data without discarding any part of the data stored in the memory space, when the transmission of the information data is interrupted according to a default setting of a network.

19. A data communication method comprising the steps of:

transmitting information data by using an address specifying a part of the memory space provided in a destination;
storing the information data in a part of the memory space specified by the address; and
resuming transmission of said information data from one part of said information data by using an address specified by said destination or said controller, when the transmission of said information data is interrupted according to a default setting of a network.

20. A data communication method comprising the steps of:

transmitting s information data by using an address specifying a part of the memory space provided in a destination; and
resuming transmission of the information data from one part of said information data by using an address specified by said destination or said controller, when the transmission of the information data is interrupted according to a default setting of a network.

21. A data communication method comprising steps of:

storing information data transmitted from a source in a part of the memory space specified by said source; and
specifying an address to said source specifying one part of the memory space, when a transmission of said information data is interrupted according to a default setting of a network.

22. A data communication method comprising the steps of:

specifying a beginning of transmission of information data, along with specifying an address that specifies one part of a memory space in relation to said source; and
specifying an address specifying one part of the memory space to said source, when a transmission of said information data is interrupted according to a default setting of a network.

23. A data communication apparatus comprising:

means for transmitting information data by using an address specifying one part of a memory space provided in a destination; and
means for executing control such that transmis-

sion of said information data is resumed without discarding any part of the data stored in the memory space, when the transmission of said information data is interrupted according to a default setting of a network.

24. A data communication apparatus comprising:

means for storing information data transmitted from a source to one part of a memory space specified by a source; and

means for executing control such a transmission of the information data is resumed without discarding any part of the data stored in the memory space, when the transmission of the information data is interrupted according to a default setting of a network.

25. A data communication apparatus comprising:

means for transmitting information data by using an address specifying one part of a memory space provided in a destination; and

means for executing control such that transmission of the information data is resumed by using an address specified by said destination or said controller, when the transmission of the information data is interrupted according to a default setting of a network.

26. A data communication apparatus comprising:

means for storing information data transmitted from a source to one part of a memory space specified by a source; and

means for specifying an address to said source that specifies one part of the memory space, when a transmission of the information data is interrupted according to a default setting of a network.

FIG. 1

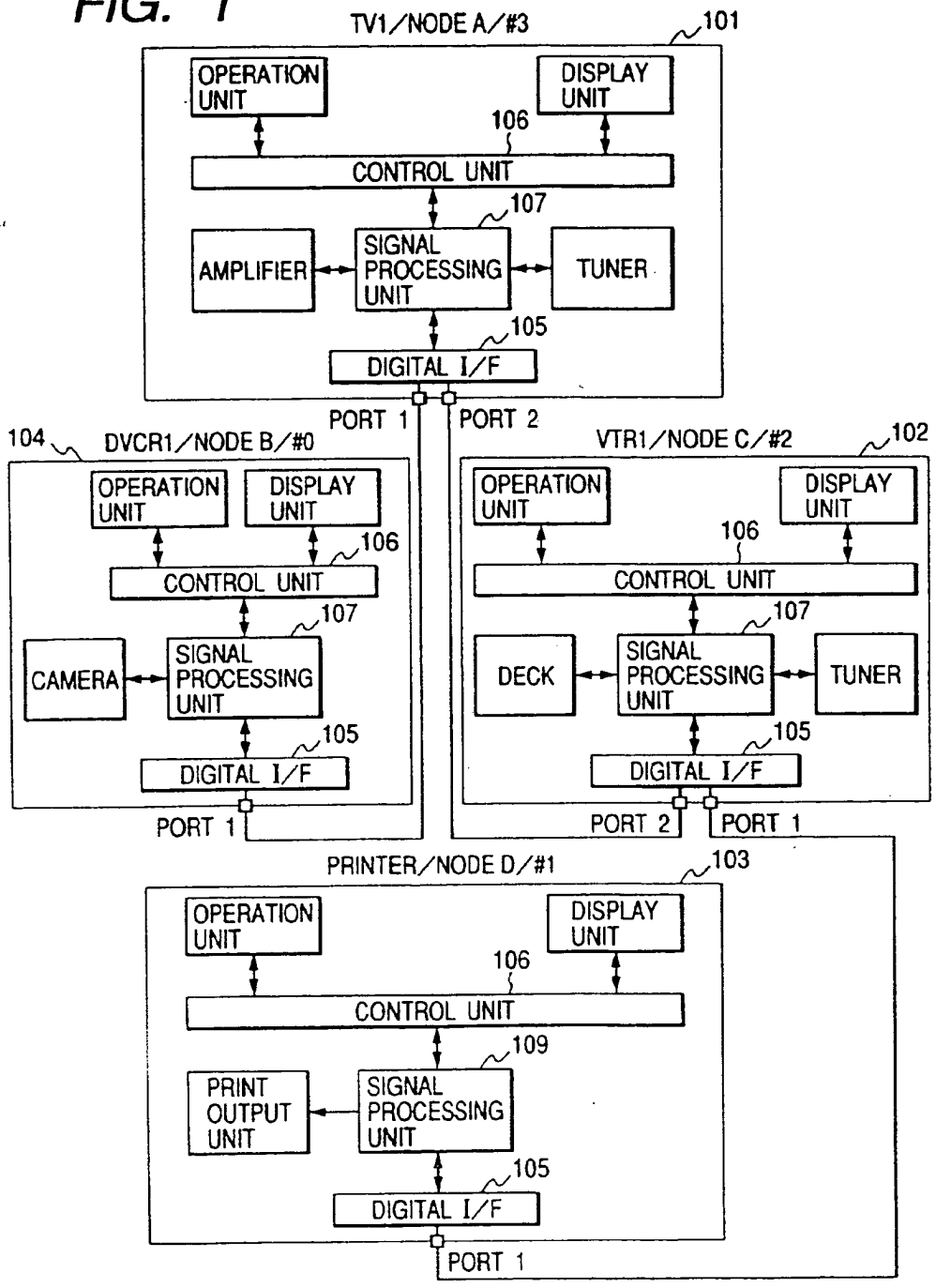

TV1/NODE A/#3
101
OPERATION UNIT
DISPLAY UNIT
106
CONTROL UNIT
107
AMPLIFIER
SIGNAL PROCESSING UNIT
TUNER
105
DIGITAL I/F
PORT 1
PORT 2
104
DVCR1/NODE B/#0
OPERATION UNIT
DISPLAY UNIT
106
CONTROL UNIT
107
CAMERA
SIGNAL PROCESSING UNIT
105
DIGITAL I/F
PORT 1
VTR1/NODE C/#2
102
OPERATION UNIT
DISPLAY UNIT
106
CONTROL UNIT
107
DECK
SIGNAL PROCESSING UNIT
TUNER
105
DIGITAL I/F
PORT 2
PORT 1
PRINTER/NODE D/#1
103
OPERATION UNIT
DISPLAY UNIT
106
CONTROL UNIT
109
PRINT OUTPUT UNIT
SIGNAL PROCESSING UNIT
105
DIGITAL I/F
PORT 1

FIG. 2

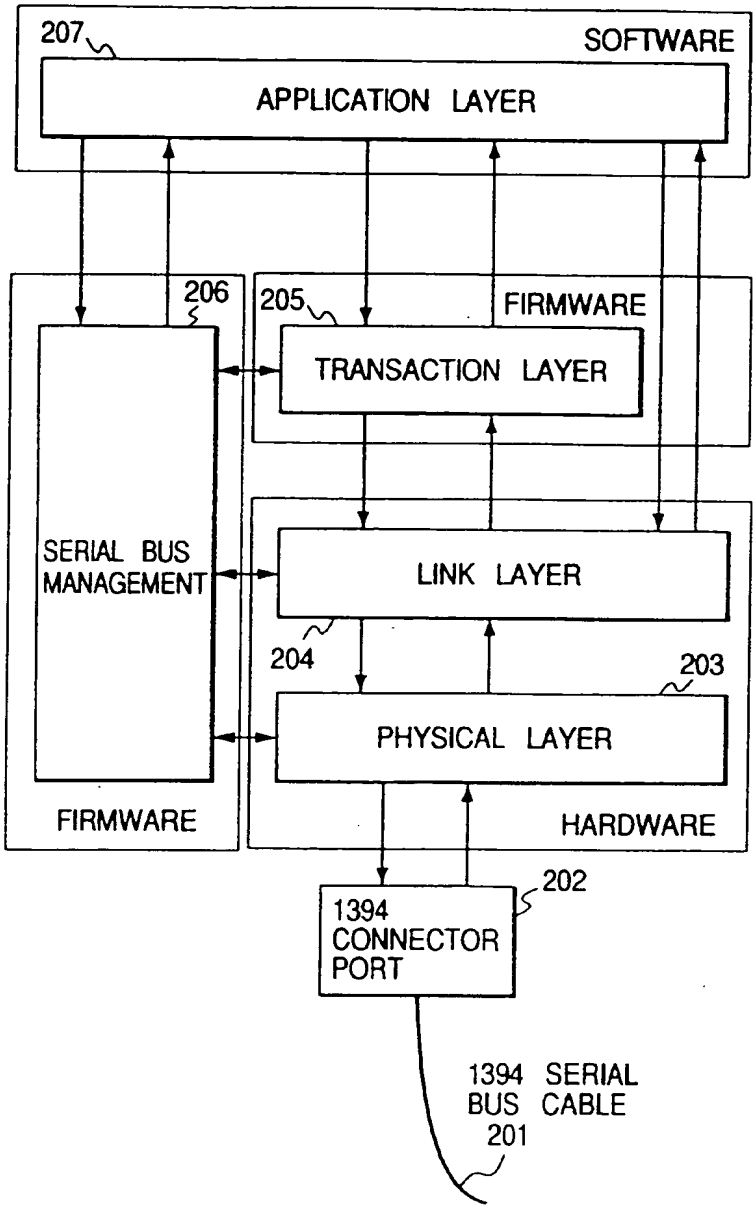

207
SOFTWARE
APPLICATION LAYER
206
205
FIRMWARE
TRANSACTION LAYER
SERIAL BUS MANAGEMENT
LINK LAYER
204
203
PHYSICAL LAYER
FIRMWARE
HARDWARE
202
1394 CONNECTOR PORT
1394 SERIAL BUS CABLE
201

FIG. 3

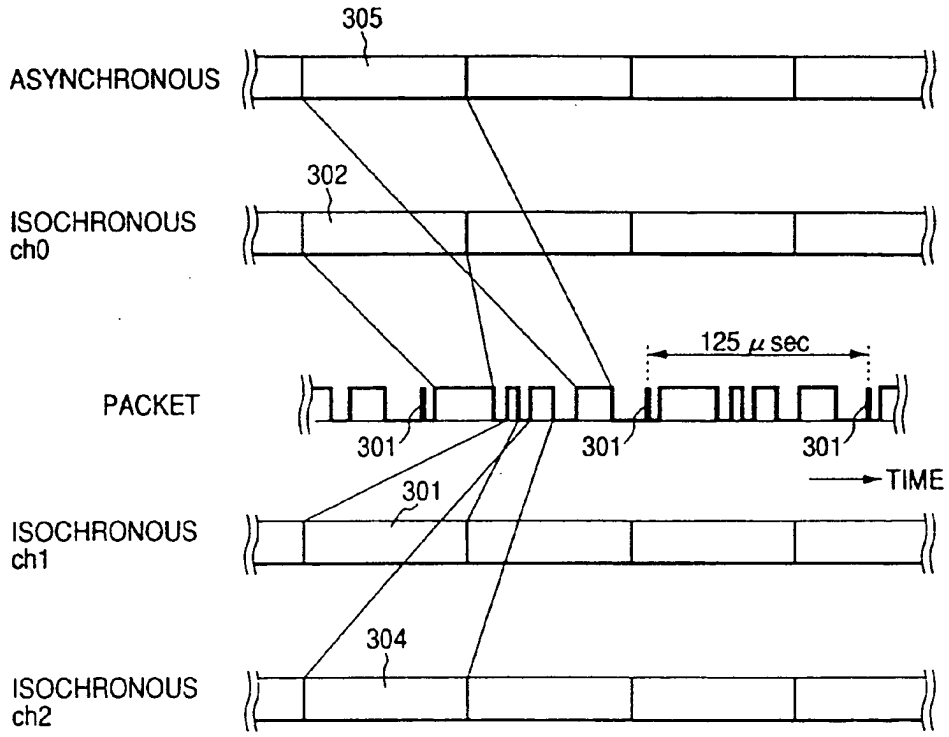

305
ASYNCHRONOUS
302
ISOCHRONOUS ch0
125 μ sec
PACKET
301
301
301
TIME
301
ISOCHRONOUS ch1
304
ISOCHRONOUS ch2

FIG. 4

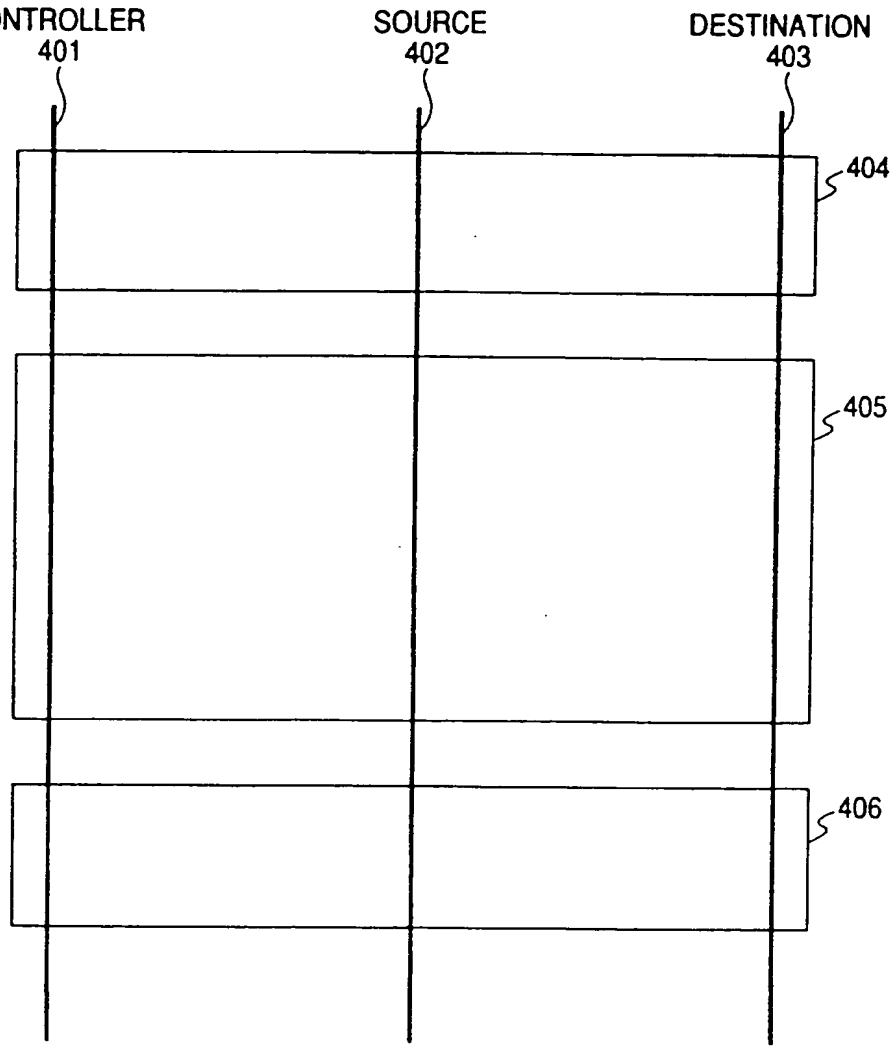


FIG. 5

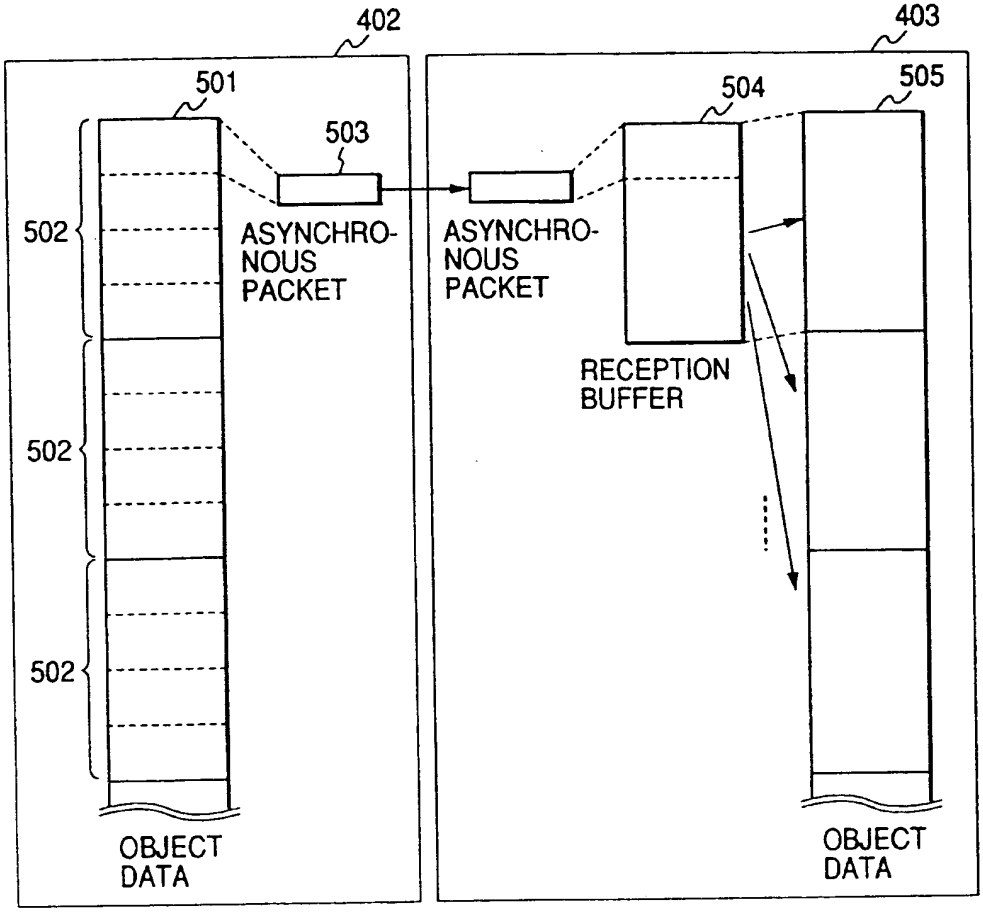

402
403
501
503
504
505
502
502
502
ASYNCHRO-
NOUS
PACKET
ASYNCHRO-
NOUS
PACKET
RECEPTION
BUFFER
OBJECT
DATA
OBJECT
DATA

FIG. 6A

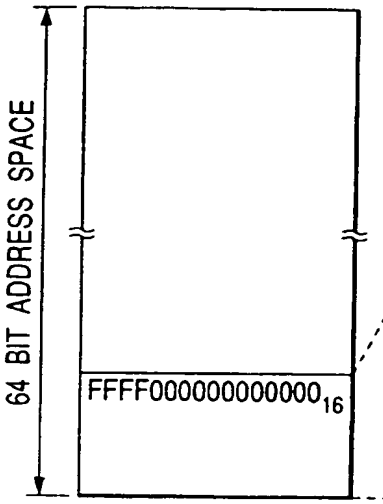


FIG. 6B

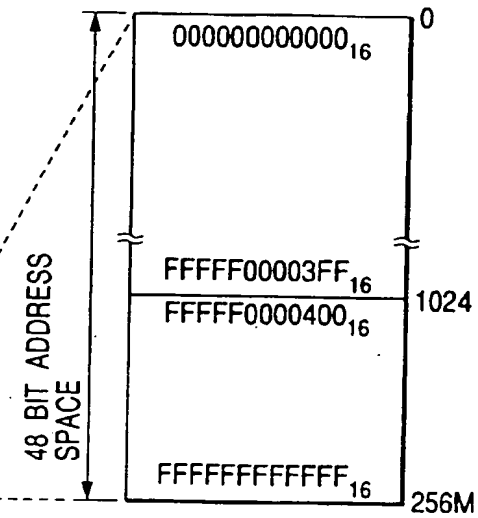


FIG. 8

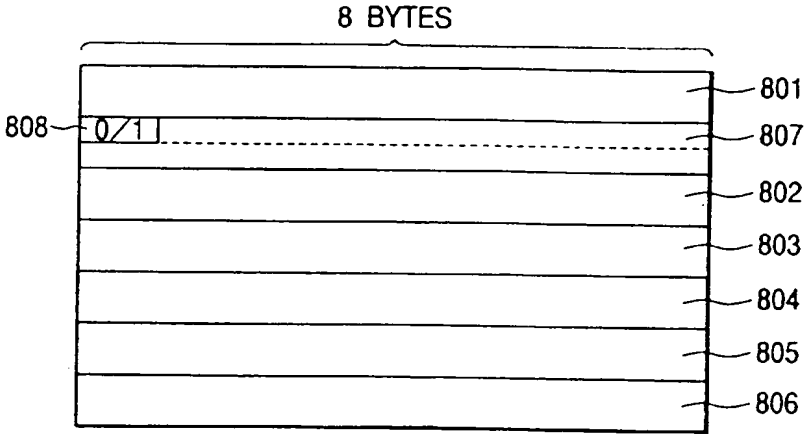


FIG. 7

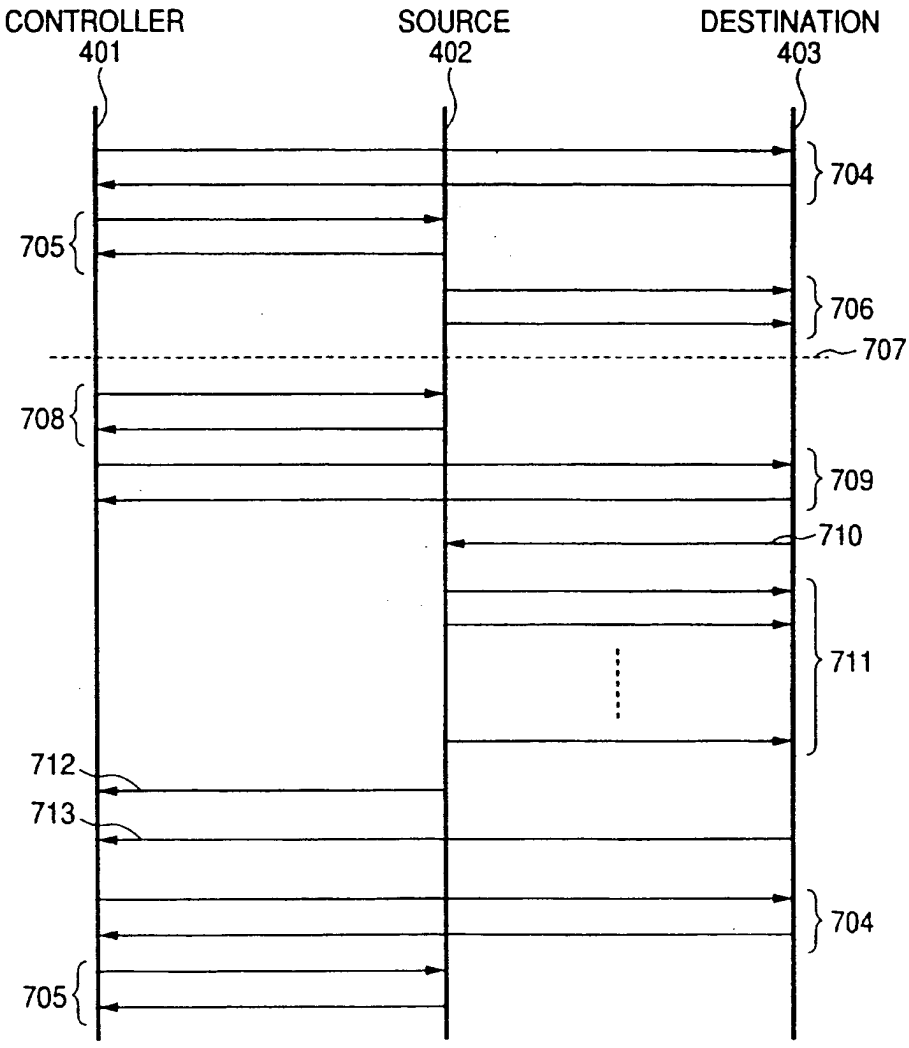

CONTROLLER
401
SOURCE
402
DESTINATION
403
704
705
706
707
708
709
710
711
712
713
704
705

FIG. 9

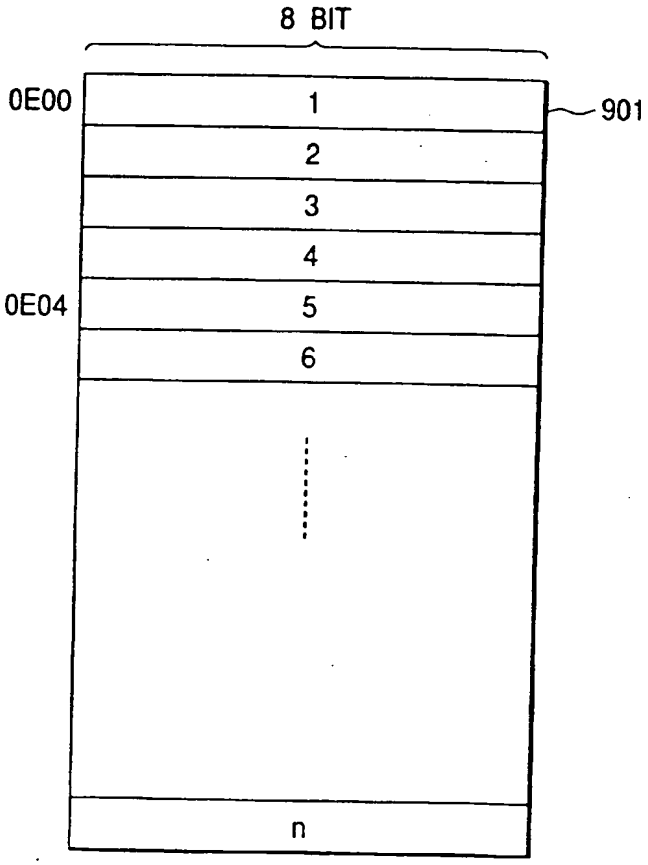


FIG. 10

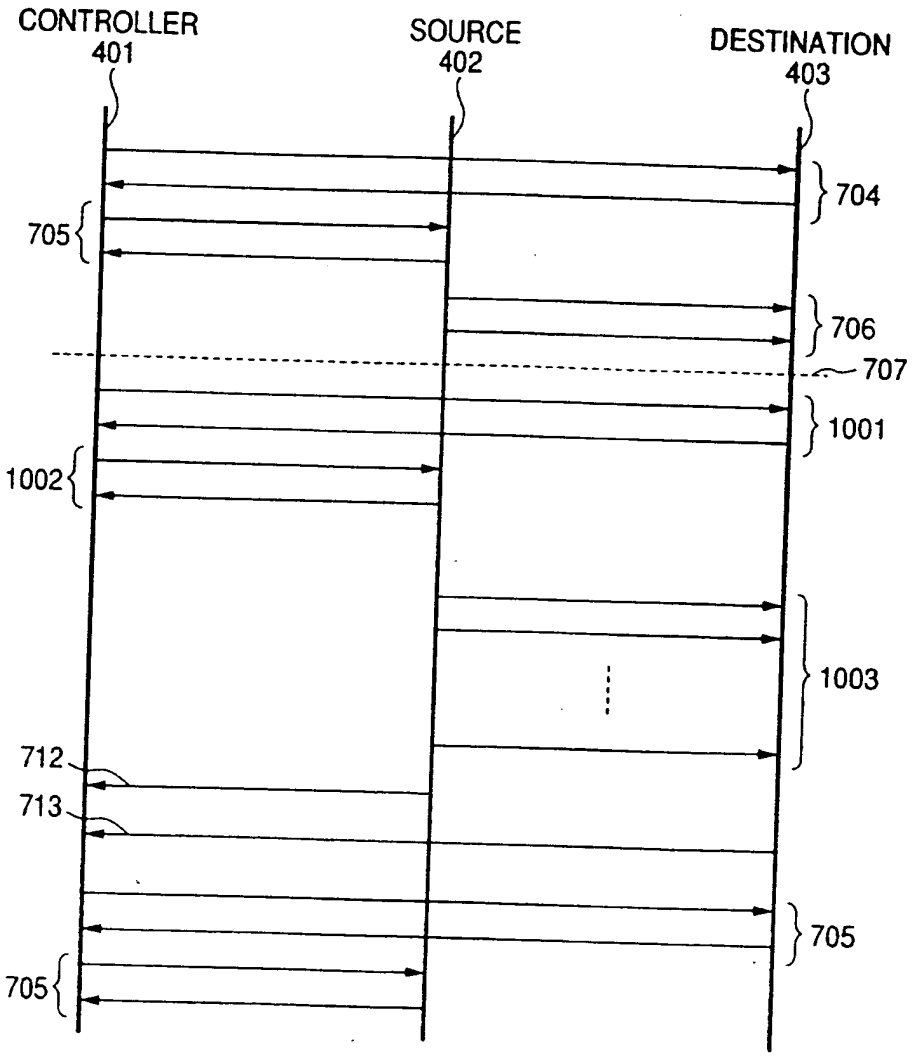

CONTROLLER
401
SOURCE
402
DESTINATION
403
704
705
706
707
1001
1002
1003
712
713
705
705

FIG. 11

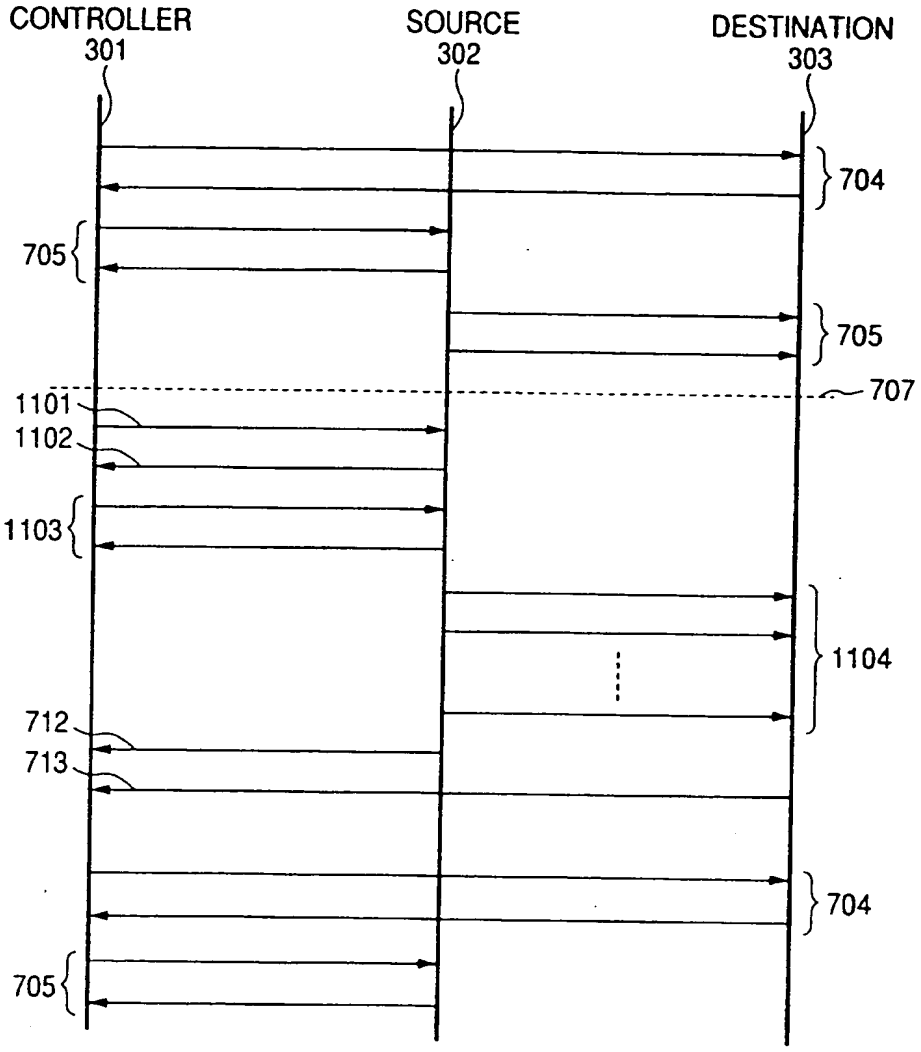
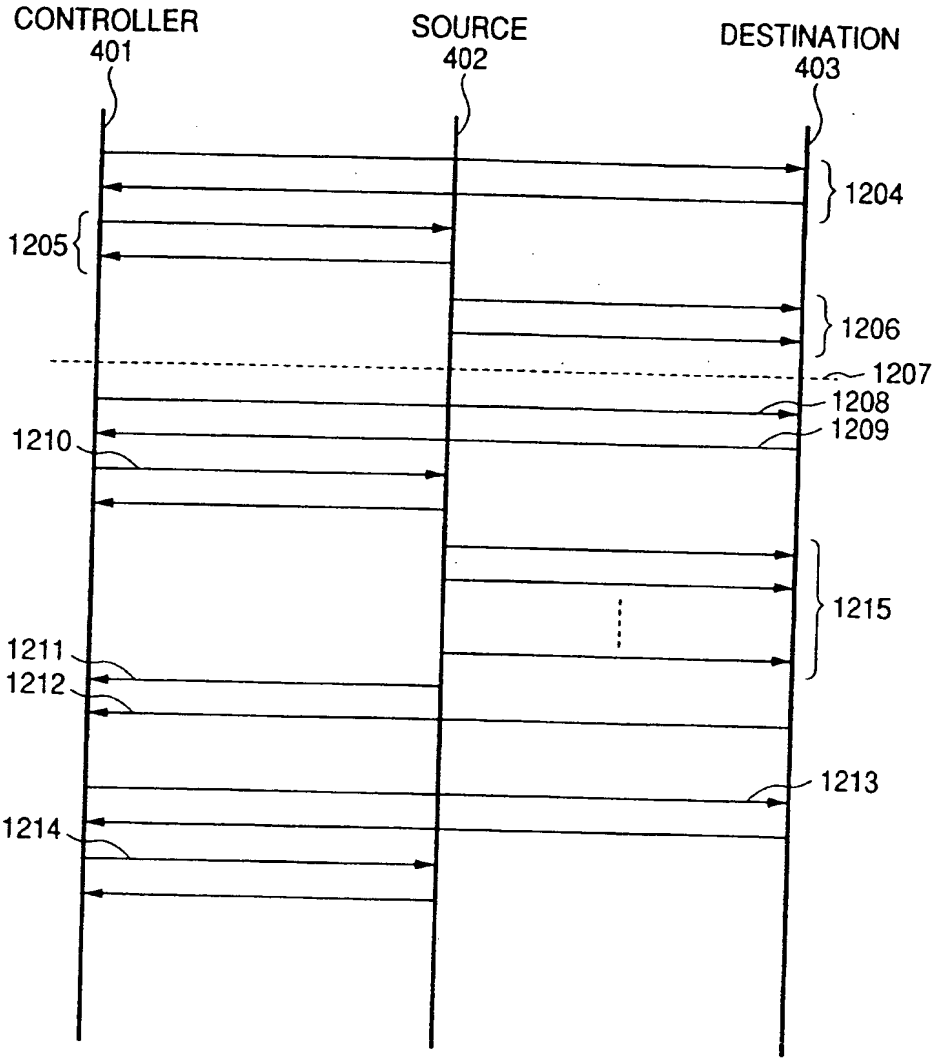

CONTROLLER
301
SOURCE
302
DESTINATION
303
704
705
705
707
1101
1102
1103
1104
712
713
704
705

FIG. 12


CONTROLLER
401
SOURCE
402
DESTINATION
403
1204
1205
1206
1207
1208
1209
1210
1215
1211
1212
1213
1214